A Low Control Overhead Cluster Maintenance Scheme for Mobile Ad hoc NETworks (MANETs)

Narendra Singh Yadav [1], Bhaskar P Deosarkar [2], and R.P.Yadav[3]
Department of Electronics & Communication Engineering,
Malaviya National Institute of Technology,
Jaipur, India
Email: {narensinghyadav@yahoo.com, bhaskar44_nanded@yahoo.co.in, rp_yadav@yahoo.com}

***Abstract*— Clustering is an important research area for mobile ad hoc networks (MANETs) as it increases the capacity of network, reduces the routing overhead and makes the network more scalable in the presence of both high mobility and a large number of mobile nodes. In clustering the clusterhead manage and store recent routing information. However the frequent change of clusterhead leads to loss of routing information stored, changes the route between two nodes, affects the performance of the routing protocol and makes the cluster structure unstable. Communication overhead in terms of exchanging messages is needed to elect a new clusterhead. The goal then would be to keep the clusterhead change as least as possible to make cluster structure more stable, to prevent loss of routing information which in turn improve the performance of routing protocol based on clustering. This can be achieved by an efficient cluster maintenance scheme. In this work, a novel clustering algorithm, namely Incremental Maintenance Clustering Scheme (IMS) is proposed for Mobile Ad Hoc Networks. The goals are yielding low number of clusterhead and clustermember changes, maintaining stable clusters, minimizing the number of clustering overhead. Through simulations the performance of IMS is compared with that of least cluster change (LCC) and maintenance scheme of Cluster Based Routing Protocol (CBRP) in terms of the number of clusterhead changes, number of cluster-member changes and clustering overhead by varying mobility and speed. The simulation results demonstrate the superiority of IMS over LCC and maintenance scheme of CBRP.**

***Index Terms*—MANET, CBRP, cluster maintenance, control overhead, routing**

I. Introduction

With the increase of small size information processing devices, like laptop, pocket PC and PDA, the growing need to exchange digital information among people within a short communication range caused the emergence of Mobile Ad hoc NETworks (MANETs). MANET can be defined as an autonomous system of mobile nodes connected by wireless links, in which the nodes organize themselves arbitrarily and are free to move randomly. Some of their most interesting features are the possibility of multi-hop communication, the lack of a fixed centralized infrastructure and capability of self-organization. These features made them attractive for battlefield, emergency operations such as search and rescue in which the deployment of a fixed infrastructure can be costly, risky, and time-consuming. MANETs are characterized by their distributed nature of operation, dynamic topology, limited physical security, peer-to-peer nature, utilization of multihop relaying and dependency on battery life. The major function of any data network is to transport the data, from the intended source to the desired destination, in a reliable way and in a minimum time. But the error prone nature and limited range of the shared wireless medium, absence of any fixed infrastructure, limited resources of the mobile nodes and the dynamic topology impose certain restrictions on establishing and maintaining the routes for such a data delivery and make the task of routing and resource management difficult and challenging. Several protocols and architectures [1] [2] are proposed in the literature for performing this task which may broadly classified as flat and hierarchical based on the topological arrangement of nodes assumed.

To meet the expected demand of allowing the new nodes to join the network and existing nodes to leave the network, the architecture used should have sufficient scalability. But it is proved earlier that flat architecture is not as much scalable [3][4][5]. Clustering is proved to be scalable and bandwidth efficient structure which basically forms a hierarchical arrangement of the nodes [6]. The basic purpose behind clustering is to form and maintain a connected cluster structure. It consist of two phases- the cluster formation which deals with building of cluster structure and the cluster maintenance that deals with updating the cluster structure according to the changing network topology and is quite important being related to the performance of the given clustering algorithm.

The clusterhead manages and stores recent routing information. Communication overhead in terms of exchanging messages is needed to elect a new clusterhead. The frequent change of clusterhead makes the cluster unstable due to loss of routing information which may cause the change in the route between two nodes and hence affecting the performance of the routing protocol. The goal then would be to keep the clusterhead change as least as possible, to make cluster structure more stable, to prevent loss of routing information which in turn improve the performance of routing protocol. This can be achieved by an efficient cluster maintenance scheme. The basic idea behind this is to delay clusterhead change when two clusterheads are within the range of each other so that the passing clusterhead moves out of the range of other clusterhead to avoid unnecessary clusterhead change.

Accordingly various cluster maintenance schemes are proposed in the literature in which when two clusterhead

DOI: 01.ijns.01.01.01

come into range of each other one of the clusterhead leaves its role based on certain predefined criterion. Particularly, in LCC [7] the criterion used may be Lowest ID or Highest Connectivity based on the clustering scheme used and the clusterhead change is delayed by a predefined time called as “contention period” in CBRP [8]. But these schemes may impose an unnecessary clusterhead change when two clusterheads are just passing each other. To avoid such unwanted clusterhead change an Incremental cluster Maintenance Scheme (IMS) is proposed in this paper.

Rest of the paper is organized as follows. In section II, LCC and cluster maintenance in CBRP are explained in brief as are related to the scheme proposed in this paper. The proposed Incremental Maintenance Scheme (IMS) is presented in section III. Section IV explains the simulation parameters and performance metrics used. Results are presented and discussed in Section V. Finally Section VI concludes this paper.

II. Related Works

In this section previously proposed cluster maintenance schemes are discussed in brief.

A. *Least clusterhead change [LCC]*

In previous work, two simple criterions are used to form the clusters: One based on node ID and the other based on node degree. Specifically, lowest-ID nodes (LIC [9]) and maximum-degree nodes (HCC [10]) respectively are elected to be the clusterheads in cluster formation.

In LIC, the node with the lowest ID among its neighbors is elected as clusterhead. When a clusterhead finds a member with ID lower than its own ID then the clusterhead is forced to handover the clusterhead role to this node with lowest ID.

In HCC, the clustering is performed periodically to check the “local highest node degree” attribute of a clusterhead. When a clusterhead finds a member node with a higher degree, it is forced to relinquish its clusterhead role.

Both LIC and HCC mechanism involves frequent re-clustering due to mobility of the nodes. To avoid frequent re-clustering in these schemes an improvement is suggested by LCC.

In LCC the clustering algorithm is divided into two steps: cluster formation and cluster maintenance. The cluster formation simply follows LIC, i.e. initially mobile nodes with the lowest ID in their neighborhoods are chosen as clusterheads. Re-clustering is event-driven and invoked in only two cases:

- When two clusterheads are within radio range of each other the clusterhead with lowest ID continues to work as clusterhead and forcing other to relinquish its role. A simple member node is not allowed to challenge the clusterhead even if it has an ID lower to clusterhead.
- When a node cannot access any clusterhead, it rebuilds the cluster structure for the network according to LIC.

Hence, LCC significantly improves cluster stability by relinquishing the requirement that a clusterhead should always bear some specific attributes in its local area. But the second case of re-clustering in LCC indicates that a single node’s movement may still invoke the complete cluster structure re-computation, and once this happens, the large communication overhead for clustering may not be avoided. And as and when two clusterhead come into radio range of each other the clusterhead change is bought in. This is unwanted when two clusterheads are just passing each other and may be in each others range for a short duration.

B. *Cluster maintenance in CBRP:*

In CBRP, as clusters are identified by their respective cluster heads it is desirable to have clusterhead changes as minimum as possible. For maintaining the cluster following clusterhead change rules are imposed in CBRP, as described in [8].

- A non-cluster head never challenges the status of an existing cluster head.
- When two cluster heads move next to each other over an extended period of time (for CONTENTION_PERIOD seconds), then only will one of them lose its role of cluster head.

As a result, whenever a cluster head hears HELLO messages from another cluster head indicating a bi-directional link, it sets c_timer to expire in CONTENTION_PERIOD seconds. When c_timer expires, it will check if it is still in contention with the other cluster head, by checking if the other cluster head is still in its neighbor table. If so, it compares its own ID with that of the other cluster head's. The one with a smaller ID will continue to act as cluster head. The one with a bigger ID gives up its role as cluster head and changes from C_HEAD to C_MEMBER in its subsequent HELLO messages. This might trigger reorganization of other clusters. These rules guarantee some sort cluster stability by delaying the clusterhead change by CONTENTION_PERIOD upon coming of two clusterheads in each others range. This avoids unnecessary clusterhead change if their passing time is less than or equal to CONTENTION_PERIOD but if passing time is more the clusterhead change is forced.

III. Proposed Incremental Maintenance Scheme [IMS]

The cluster formation mechanism in IMS is based on lowest ID clustering algorithm in which the node with lowest ID in neighborhood is elected as a clusterhead. In the proposed scheme, when two clusterheads are with in range of each other, clusterhead change is delayed for delay_period which is equal to Hello_interval initially. If after delay_period both are again with in range of each other then delay_period is increased by Hello_interval. Delay_period is incremented by Hello_interval every time both clusterheads are within range of each other, till delay_period is less than or equal to max_limit which is obtained by dividing two times transmission range by speed. If both are still with in range then the one with a smaller ID will continue to act as a clusterhead and the other one gives up its role as clusterhead as shown in following Fig. 1.

DOI: 01.ijns.01.01.01

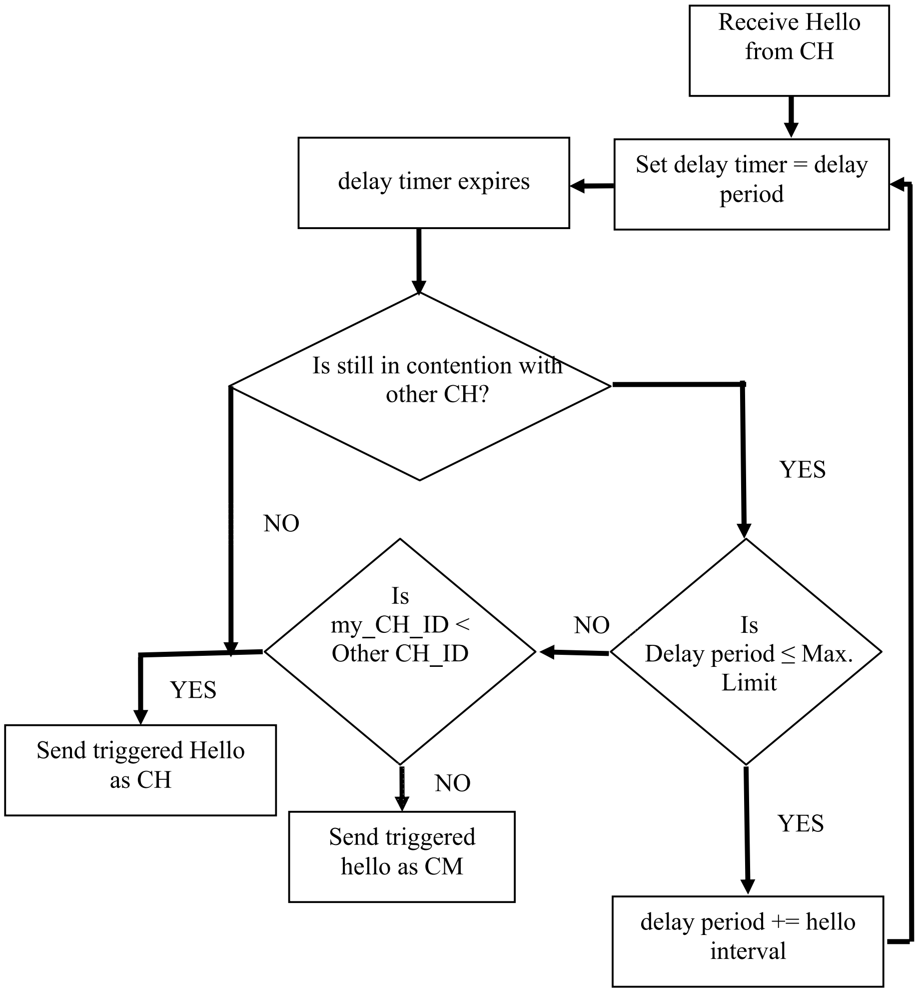

Figure 1. Flowchart depicting the clusterhead change in proposed scheme

IV. Simulation Parameters and Performance Metric

LCC, cluster maintenance scheme in CBRP and the proposed IMS are implemented in NS2 [11]. It is an object-oriented, discrete event driven network simulator developed at UC Berkely written in C++ and OTcl and is particularly popular in the ad hoc networking research community. In this simulation study the source-destination pairs are spread randomly over the network. The node movement generator of NS-2 is used to generate the different node movement scenarios. The node movement is assumed to follow the random way point model. The movement generator takes the number of nodes, pause time, maximum speed, field configuration and simulation time as input parameters. The propagation model used is two ray ground [12]. Simulations consist of two stages. In stage1 simulations are carried out by varying the mobility (pause time) and in stage2 by varying the node speed. The simulation parameters used are listed in Table1. Five runs of each scenario are simulated for 300 seconds to collect the desired data at steady state to obtain statistically confident averages.

TABLE 1
SIMULATION PARAMETERS

Parameters	Stage I	Stage II
Number of mobile nodes, N	150	
Simulation Area	2000m x 500m	
Simulation Time	300s	
Pause time for mobile nodes	0, 60, 120, 180, 240 and 300s	100 s
Max. speed for mobile nodes	10m/s	5, 10, 15, 20 and 25 m/s
Transmission range for mobile nodes	250m	

Performance Metrics: The metrics considered for evaluations are the number of clusterhead change, the number of cluster member change and clustering overhead by varying pause time and speed.

The number of clusterhead change is the total number of clusterhead changes during the whole simulation run

DOI: 01.ijns.01.01.01

time. A small value of clusterhead change reflects the stability of the cluster structure.

The number of cluster member change is the number of mobile nodes that switch to another clusterhead during the simulation run time.

Clustering overhead is the number of clustering messages sent by each node in cluster formation and cluster maintenance operation. It is an important measure for the scalability of a protocol.

V. RESULTS AND DISCUSSION

Fig.2 - Fig. 4 shows the performance of IMS, CBRP and LCC in terms of number of clusterhead changes, number of cluster member changes and clustering overhead as function of pause time and performance with respect to speed is shown in Fig. 5 – Fig. 7 The pause time is varied from 0 sec to 300 sec in steps of 60 sec and number of clusterhead changes, cluster member changes and clustering overhead is observed. For observing the effect of speed the node speed is varied from 5 m/s to 25 m/s in steps of 5 m/s and the performance metric is evaluated.

Fig. 2 shows the number of (#) clusterhead changes as a function of pause time. In LCC, CBRP and IMS all as pause time increases the required number of clusterhead changes are very low. From figure it is clear that IMS performs better to LCC and CBRP both. At highest mobility when the nodes are continuously moving the number of clusterhead changes required in IMS are approximately one third that of LCC and half of CBRP. Pause time upto 180 sec CBRP scheme performs slightly better to LCC after which both perform similarly whereas IMS outperforms both of them. This difference in performance is due to the clusterhead change delay strategy used in CBRP and IMS. In CBRP the delay is of the time equal to CONTENTION_PERIOD whereas in IMS it is a function of speed and transmission range.

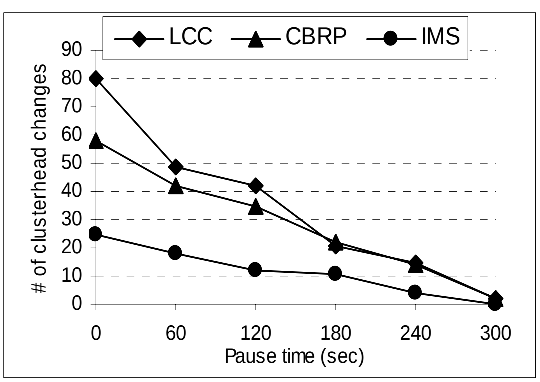

Figure 2 Number of clusterhead change vs. pause time

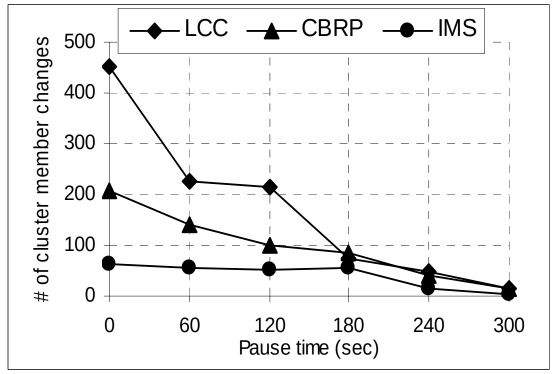

Figure 3 Number of cluster member change vs. pause time

Fig. 3 shows the number of cluster member changes as a function of pause time. In LCC, CBRP and IMS all as pause time increases the required number of cluster member changes are very low. From figure it is clear that IMS performs better to LCC and CBRP both. At highest mobility when the nodes are continuously moving the number of cluster member changes required in IMS are approximately one ninth of that of LCC and fourth of CBRP. Pause time upto 180 sec CBRP scheme performs better to LCC after which both perform similarly whereas IMS outperforms both of them. This change in behavior can be reasoned to the number of clusterhead changes required in respective schemes. The less number of clusterhead changes indicate the reduced number of re-affiliations in IMS as compared to LCC and CBRP schemes.

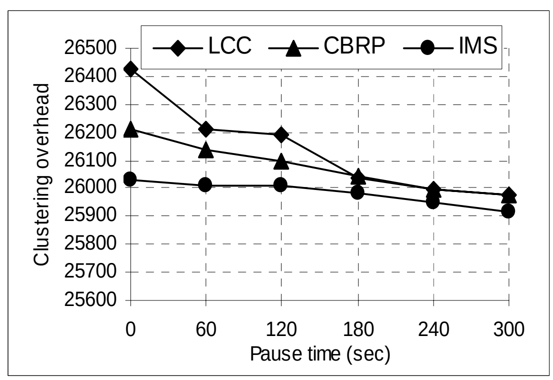

Figure 4 clustering overhead vs. pause time

Fig. 4 shows the clustering overhead, in LCC, CBRP and IMS, as a function of pause time. From figure it can be observed that the clustering overhead required in IMS are comparatively less than LCC and CBRP. In IMS the less number of clusterhead changes triggers less number of cluster member re-affiliations which are the main sources of control overhead.

Fig.5 below shows variation in clusterhead changes with respect to variation in speed. In all three schemes LCC, CBRP and IMS, the number of clusterhead changes increases with increase in speed. The number of clusterhead changes in IMS at all the speeds between 5

DOI: 01.ijns.01.01.01

ACEEE

m/s and 25 m/s are very less as compared to LCC and CBRP. The sudden increase in number of clusterhead changes at 20m/s may be due to simulation restrictions.

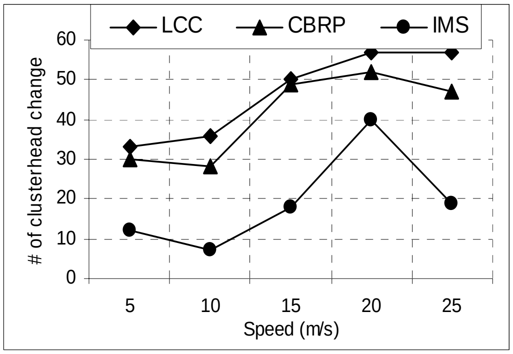

Figure 5 Number of clusterhead change vs. speed

Fig. 6 shows variation in cluster member changes with respect to variation in speed. In all three schemes LCC, CBRP and IMS, the number of cluster member changes increases with increase in speed. The less number of clusterhead changes in IMS causes less cluster member re-affiliations and hence the number of cluster member changes required in IMS are very less than that of LCC and CBRP.

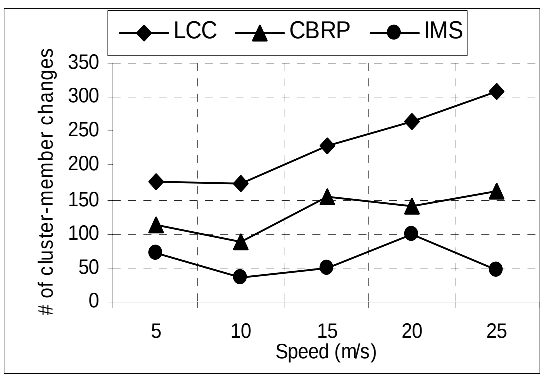

Figure 6 Number of cluster member change vs. speed

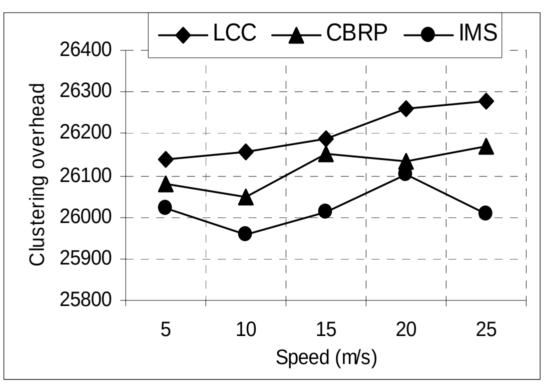

Figure 7 clustering overhead vs. speed

Fig. 7 shows the clustering overhead, in LCC, CBRP and IMS, as a function of speed. From figure it can be observed that the clustering overhead required in IMS are comparatively less than LCC and CBRP. In IMS the less number of clusterhead changes triggers less number of cluster member re-affiliations which are the main sources of control overhead.

VI. Conclusion

Since the stability of cluster in a cluster based Mobile Ad hoc networks affects the performance of protocols such as scheduling, routing and signaling, a clustering maintenance scheme, named IMS, had been proposed and studied in this paper. The basic idea of this scheme is to delay clusterhead change when two clusterheads are within transmission range of each other to avoid unnecessary clusterhead change. Simulation results show that IMS is better cluster maintenance scheme as compared to LCC and the scheme used in CBRP in terms of number of clusterhead changes, number of cluster member changes and clustering overhead. This is due to avoiding the unnecessary clusterhead changes. In conclusion, IMS successfully fulfills its aim of providing a stable cluster structure for MANETs

References

[1] M. Abolhasan, T. Wysocki, A. Dutkiewicz, “A review of routing protocols for mobile ad hoc networks”, Elsevier Science, Journal of Ad Hoc Networks, Vol.2, 2004, pp.1-22

[2] E. M. Royer and C. K. Toh, “A review of current routing protocols for ad hoc mobile wireless networks,” IEEE Personal Communications magazine, April 1999, pp. 46–55.

[3] P. Gupta and P. R. Kumar, “The Capacity of Wireless Networks,” IEEE Trans. Info. Theory, vol- IT 46.2, Mar. 2000, pp. 388-404.

[4] X. Y. Hong, K. X. Xu and M. Gerla, “Scalable Routing Protocols for Mobile Ad Hoc Networks,” IEEE Network, July-Aug. 2002, pp. 11-21.

[5] X. Y. Hong, K. X. Xu and M. Gerla, “An Ad Hoc Network with Mobile Backbones,” Proc. IEEE ICC 2002, vol. 5, Apr.-May 2002, pp. 3138-43

[6] E. M. Belding-Royer, “Hierarchical Routing in Ad Hoc Mobile Networks,” Wireless Commun. And Mobile Comp., vol. 2, no. 5, 2002, pp. 515-32.

[7] C.-C. Chiang, H.-K. Wu, W. Liu and M. Gerla, “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,” in *Proc. IEEE SICON’97*, 1997.

[8] Mingliang Jiang, Jinyang Li and Y.C.Tay, “Cluster Based Routing Protocol”, August 1999 IETF Draft. http://www.ietf.org/internet-drafts/draft-ietf-manetcbrp-spec-01.txt

[9] A. Ephremides, J. E. Wieselthier, and D. J. Baker, “A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling,” in *Proc. IEEE*, vol. 75, 1987, pp. 56–73.

[10] M. Gerla and J. T. Tsai, “Multiuser, Mobile, Multimedia Radio Network,” *Wireless Networks*, vol. 1, Oct. 1995, pp. 255–65.

[11] K. Fall and K. Vardhan, The Network Simulator (ns-2). Available: http://www.isi.edu/nsnam/ns

[12] T. S. Rappaport, Wireless Communications, Principles & Practices. Prentice Hall, 1996, ch. 3, pp. 70-74.

DOI: 01.ijns.01.01.01